Read and Write Voltage Signal Optimization for Multi-Level-Cell (MLC) NAND Flash Memory

Chaudhry Adnan Aslam, *Student Member, IEEE,* Yong Liang Guan, *Member, IEEE,*
Kui Cai , *Senior Member, IEEE*

***Abstract*—The multi-level-cell (MLC) NAND flash channel exhibits non-stationary behavior over increasing program and erase (PE) cycles and data retention time. In this paper, an optimization scheme for adjusting the read (quantized) and write (verify) voltage levels to adapt to the non-stationary flash channel is presented. Using a model-based approach to represent the flash channel, incorporating the programming noise, random telegraph noise (RTN), data retention noise and cell-to-cell interference as major signal degradation components, the write-voltage levels are optimized by minimizing the channel error probability. Moreover, for selecting the quantization levels for the read-voltage to facilitate soft LDPC decoding, an entropy-based function is introduced by which the voltage erasure regions (error dominating regions) are controlled to produce the lowest bit/frame error probability. The proposed write and read voltage optimization schemes not only minimize the error probability throughout the operational lifetime of flash memory, but also improve the decoding convergence speed. Finally, to minimize the number of read-voltage quantization levels while ensuring LDPC decoder convergence, the extrinsic information transfer (EXIT) analysis is performed over the MLC flash channel.**

***Index Terms*—MLC NAND flash memory, read-voltage, write-voltage, LDPC code, error performance.**

I. Introduction

THE NAND flash memory is the ubiquitous storage medium in many consumer electronic products. With multi-level-cell (MLC) technology, flash memory can store multiple bits over a single memory cell, leading to significant growth in its storage capacity. Using advanced chip manufacturing processes, it has become viable to replace magnetic storage disks with NAND flash memory based solid-state drives (SSD) for large enterprise data applications.

In flash memory, a cell can either be represented with the *erased* state (no electrons stored over the floating-gate) or with the *programmed* state (electrons stored over the floating-gate). Initially, all memory cells are in the erased state. The programming operation shifts the threshold voltage (voltage required to turn-on the transistor) of a memory cell to a particular write-voltage level. Considering an example of $2-$bit per cell flash memory, a memory cell can be configured to four distinct voltage levels, say $V_{\min}, V_1, V_2, V_{\max}$, for representing data symbols '11', '10', '00'and '01', respectively, where $V_{\min}$ is the mean threshold voltage of the erased cells and $V_{\max}$ is the mean threshold voltage of programmed cells configured with the highest write-voltage level. It is observed that the probability distribution functions for these distinct voltage levels are not identical due to the presence of some non-stationary and asymmetric channel noise. In this scenario, $V_{\min}$, V_1, V_2 and $V_{\max}$ should not be equally-spaced but determined based on some optimization criteria.

To read a flash memory cell, the amount of stored electrical charge is measured by applying the read-voltage in discrete steps. For a memory cell configured to threshold voltage V_{th}, a read-voltage greater than V_{th} is required to measure the stored electric charge. For $2-$bit per cell flash memory, we need at least 3 read-voltage levels R_1, R_2 and R_3, where R_1 can be set between $V_{\min}$ and V_1 to distinguish between symbols '11'and '10', R_2 can be set between V_1 and V_2 to distinguish between symbols '10'and '00'and R_3 can be set between V_2 and $V_{\max}$ to distinguish between symbols '00'and '01', as shown in Fig. 1. Since the write-voltage levels are not equally-spaced, the read-voltage levels should also be appropriately designed so that the error-rate is minimized.

For a state-of-the-art flash memory, only 3-level memory sensing scheme may not be sufficient to achieve the satisfactory error performance. This is because the modern flash memory chips experience severe voltage level distortions due to some circuit-level noise and interference effects. As a consequence, flash data reliability is degraded [1]–[3]. To overcome the reliability issues, strong error correcting codes (ECCs), such as low-density parity-check (LDPC) codes, are seriously considered [4], [5]. To reap full benefits of LDPC code, it is desirable to have LDPC decoder's input values quantized as finely as possible. This requires high-precision memory sensing, and consequently more reading latency, which may not be acceptable for time-critical applications. Against these background, we analyze two research problems associated with flash memory read-voltage signal design; identifying the minimum number of read-voltage levels required under the given flash channel condition, and then finding the ideal values for these read-voltage levels.

A. Related Work

To the best of our knowledge, the write-voltage optimization over a non-stationary MLC flash channel has not been reported in the open literature. In [6], the write-voltage levels are optimized based on a simple flash channel by considering random telegraph noise (RTN) as the only source of voltage signal degradation. Another relevant work is reported in [7] where the write-voltage levels are optimized assuming an AWGN

Chaudhry Adnan Aslam and Yong Liang Guan are with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore (e-mail: ad0001ry@e.ntu.edu.sg;EYLGuan@ntu.edu.sg).

Kui Cai is with Department of Science, Singapore University of Technology and Design (email: cai_kui@sutd.edu.sg).

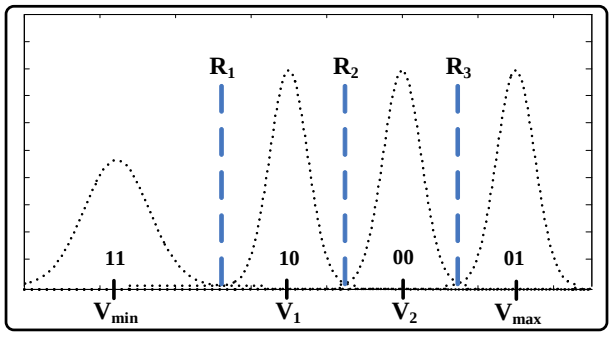

Fig. 1: Illustration of four *write-voltage* levels ($V_{\min}$, V_1, V_2 and $V_{\max}$) and three *read-voltage* levels (R_1, R_2 and R_3) for 2-bit per cell flash memory.

channel for flash memory. Our paper presents the optimization of write-voltage signals using a more comprehensive flash channel model that includes the effects of programming noise, cell-to-cell interference as well as non-stationary RTN and data retention noise.

In [8], an ECC scheme is jointly designed with write-voltage levels, relating the code's error correction capability with the voltage signal magnitude. A constrained coding scheme is presented in [9] for mitigating the effect of cell-to-cell interference in which certain voltage levels of neighboring cells are forbidden so that a wider gap between adjacent voltage distributions is assured.

In addition to MLC technology, where discrete write-voltage levels are used to store information, relative value (ordering) of charge levels can also be used to represent flash data, as discussed in *rank modulation* scheme [10]. In this scheme, non-overlapping voltage regions are required to represent data symbols. In [11], these optimal continuous voltage regions are designed. The rank modulation scheme can increase the cell storage capacity and overcome the programming overshoot errors. However, to implement this scheme, a large number of read operations are required to learn the voltage-level ordering. Thus, in this paper we focus on MLC technology and strive to optimize its discrete write-voltage levels.

In the conventional MLC flash, to avoid the effect of voltage overshoot errors, memory cells are configured on desired write-voltage levels in multiple rounds of programming. In [11]–[13], for both MLC and rank modulation schemes, the optimal programming step size has been investigated, keeping the number of programming rounds constant and assuming fixed write-voltage levels. In this paper, we address the problem of finding the optimized write-voltage levels to ensure that the flash channel error probability is minimized.

With regards to the read-voltage quantization for soft-decision decoding, multiple memory read operations are typically performed. In this direction, a simplistic approach is to apply equally spaced read-voltage levels (uniform memory sensing). However, this is not suitable for flash channel and yields poor error performance [14]. For this reason, a non-uniform quantization scheme is adopted in [14], where the quantization levels are obtained at the intersecting region between two adjacent distribution functions by using constant ratio method. Alternative to enhanced precision, hard-decision based dynamic quantization schemes are also reported in [15], [16] where the read-voltage levels are adjusted according to the non-stationary behavior of flash channel.

Another important work related to quantization design is presented in [17] in which the quantization levels are obtained by maximizing the mutual-information (MMI) between flash channel's input and output voltage signals. In this paper, we propose a read-signal quantization scheme based on a novel voltage entropy function that is able to give better decoding error performance than the MMI scheme.

B. Contributions

In this paper, we present a novel analytical approach to optimize the write-voltage signals for $2-$bit per cell flash memory. The proposed principles can be readily extended to $3-$bit or $4-$bit per cell flash technology. In view of the varying flash memory channel, which causes the threshold voltage distribution function to change over the number of PE cycles and data retention time (the duration of time since the memory cell was last programmed), we propose to adapt the write-voltage signals on-the-fly such that the channel error probability is minimized. Furthermore, we present a voltage entropy-based quantization scheme for reading the flash memory cells, so that the dominating error regions (erasure) are enclosed within the designed quantization levels. Finally, based on the given flash channel condition, we recommend to use the EXIT curves to identify the minimum number of quantization levels required for the convergence of soft LDPC decoding.

The rest of the paper is organized as follows. In Section II, we present the flash channel model. In Section III, we formulate the probability of error expression for flash channel and find optimum write-voltage levels. In Section IV, we discuss the read-voltage signal design scheme using voltage entropy function. In Section V, we analyze the error performance of the proposed read-voltage scheme. In Section VI, we discuss the improvements in decoding convergence speed as a consequence of optimized write-voltage levels. In Section VII, we compare the impact of write-voltage optimization between the flash memory's early and end-of-retention times. In Section VIII, we describe the usage of EXIT curves for the selection of memory sensing precision for LDPC decoder convergence. In Section IX, we draw conclusions.

II. MLC Flash Memory Channel Model

We model the MLC flash memory channel by incorporating the effects of programming noise, random telegraph noise (RTN), data retention noise and cell-to-cell interference (CCI). Channel models similar to ours are extensively reported in the open literature [18]–[24].

A. Initial Threshold Voltage Distribution

In flash memory array, the threshold voltage distribution for erased cells, $p_{s_{11}}$, can be modeled with Gaussian distribution [18]–[25]. given by

$$p_{s_{11}}(x) = \frac{1}{\sigma_e\sqrt{2\pi}} e^{-\frac{(x-V_{\min})^2}{2\sigma_e^2}} \quad (1)$$

where V_{min} and σ_{e} are the mean and standard deviation of cell's threshold voltage. Using a 2-bit per cell flash memory, we represent the data symbol 11 with voltage V_{min}, and 10, 00, and 01 with three programmed voltage levels V_1, V_2 and V_{max}, respectively. To configure the memory cells on one of these programmed voltage levels, an iterative incremental step pulse programming (ISPP) [26] technique is used. This causes the voltage distribution for programmed cells to follow uniform distribution [18]–[20], [24], given by

$$p_{\text{u}_\text{p}}(x) = \begin{cases} \frac{1}{\Delta V_{\text{pp}}}, & \text{for } V_{\text{p}} \leqslant x \leqslant V_{\text{p}} + \Delta V_{\text{pp}} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where V_{p} is the desired write-voltage level and ΔV_{pp} is the programming voltage step size. In this paper, we fix the value of ΔV_{pp} to 0.3, however, it can be optimized to increase the cell storage capacity [11]–[13]. Furthermore, the programmed cells are also affected by the programming noise, p_{t}, which can be modelled using Gaussian distribution [20]–[23] with zero mean and σ_{p} standard deviation. The overall voltage distribution for programmed cells can then be represented as the convolution integral ($*$) of uniform and Gaussian distribution functions, given by

$$p_{\text{s}_\text{p}}(x) = p_{\text{u}_\text{p}}(x) * p_{\text{t}}(x) \tag{3}$$

where $p_{\text{s}_\text{p}} \in \{p_{\text{s}_{10}}, p_{\text{s}_{00}}, p_{\text{s}_{01}}\}$ for $V_{\text{p}} \in \{V_1, V_2, V_{\text{max}}\}$.

B. Cell-to-Cell Interference (CCI)

The cell-to-cell interference (CCI) is induced as a result of parasitic capacitive-coupling between the adjacent memory cells. According to [14], [18]–[20], [22], [24], [27], the threshold voltage shift due to CCI, V_{CCI}, can be given as

$$V_{\text{CCI}} = \sum_{k} \Delta V_k \gamma_k \tag{4}$$

where ΔV_k is the change in the threshold voltage of interfering (neighboring) cells due to their programming and γ_{k} is the capacitive coupling ratio. A victim cell can be interfered by three or five neighboring cells. For an *even-odd* bit-line architecture, where the even bit-lines cells are programmed before the odd bit-lines cells, there are five interfering cells for each even bit-line cell and three interfering cells for each odd bit-line cell, as shown in Fig. 2. Alternatively, in the *all* bit-line architecture, where all word-line cells are programmed simultaneously, a victim cell is interfered by three neighboring cells located on the next word-line.

According to [28], the strength of CCI can be estimated before the cell programming operation and can be removed from the desired write-voltage level using cell *pre-coding* technique. However, this technique cannot mitigate the interference effect from the erased state cells. Thus, we can still approximate the voltage distribution of programmed cells using (3). However, for the erased state cells, we model their threshold voltage using Gaussian distribution function with shifted mean $\tilde{V}_{\text{min}}$, given as

$$\tilde{V}_{\text{min}}^{\text{even}} = V_{\text{min}} + \Delta V_{\text{ave}}(2\mu_{\gamma_\text{x}} + \mu_{\gamma_\text{y}} + 2\mu_{\gamma_\text{xy}}) \tag{5}$$

$$\tilde{V}_{\text{min}}^{\text{odd}} = V_{\text{min}} + \Delta V_{\text{ave}}(\mu_{\gamma_\text{y}} + 2\mu_{\gamma_\text{xy}}) \tag{6}$$

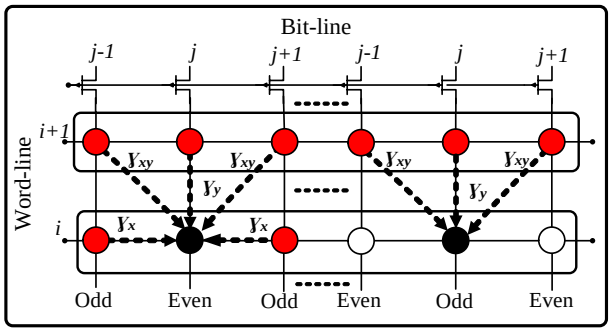

Fig. 2: Illustration of cell-to-cell interference in an *even-odd* bit-line architecture: *even* and *odd* bit-line cells are interfered by five and three neighboring cells, respectively.

where $\Delta V_{\text{ave}} = (V_{\text{min}} + V_{\text{max}})/2 - V_{\text{min}}$. Here (5) and (6) are used for even and odd bit-line cells, respectively.

C. Random Telegraph Noise (RTN)

In flash memory, the RTN is a non-stationary noise component whose effect is related with memory PE cycles[1]. According to [18]–[24], [31], it can be modeled using a symmetric exponential distribution function. However, for mathematical tractability, we approximate the RTN distribution by a zero-mean Gaussian distribution function, given by

$$p_{\text{n}}(x) = \frac{1}{\sigma_{\text{n}}\sqrt{2\pi}} e^{-\frac{x^2}{2{\sigma_{\text{n}}}^2}} \tag{7}$$

where the RTN variance, ${\sigma_{\text{n}}}^2$, is a non-stationary parameter which varies with respect to the PE cycles in a power-law fashion. In this paper, we set $\sigma_{\text{n}} = 0.00025(PE)^{0.62}$.

D. Data Retention Noise

Data retention noise is also recognized as a non-stationary and data-dependent effect related to the memory PE cycles and data retention time. According to [18]–[24], the retention noise can be approximated with Gaussian distribution, given as

$$p_{\text{r}}(x) = \frac{1}{\sigma_{\text{r}_\text{s}}\sqrt{2\pi}} e^{-\frac{(x-\mu_{\text{r}_\text{s}})^2}{2\sigma_{\text{r}_\text{s}}^2}} \tag{8}$$

We set the data-dependent mean μ_{r_s} and variance $\sigma_{\text{r}_\text{s}}^2$ parameters according to [20], as given by

$$\mu_{\text{r}_\text{s}} = (V_{\text{s}} - x_0)\,.\,[A_{\text{t}}.(PE)^{\alpha_\text{i}} + B_{\text{t}}.(PE)^{\alpha_\text{o}}]\,.\,\log(1+T) \tag{9}$$

$$\sigma_{\text{r}_\text{s}} = 0.4 \mid \mu_{\text{r}_\text{s}} \mid \tag{10}$$

where $\mu_{\text{r}_\text{s}} \in \{\mu_{\text{r}_{\text{s}_{11}}}, \mu_{\text{r}_{\text{s}_{10}}}, \mu_{\text{r}_{\text{s}_{00}}}, \mu_{\text{r}_{\text{s}_{01}}}\}$ and $\sigma_{\text{r}_\text{s}} \in \{\sigma_{\text{r}_{\text{s}_{11}}}, \sigma_{\text{r}_{\text{s}_{10}}}, \sigma_{\text{r}_{\text{s}_{00}}}, \sigma_{\text{r}_{\text{s}_{01}}}\}$ for $V_{\text{s}} \in \{V_{\text{min}}, V_1, V_2, V_{\text{max}}\}$. Here, T is the data retention time. For all our simulations, we set the following flash parameters: $V_{\text{min}} = 1.4$, $\sigma_{\text{e}} = 0.35$, $V_1 = 2.6$, $V_2 = 3.2$, $V_{\text{max}} = 3.93$, $\sigma_{\text{p}} = 0.05$, $\mu_{\gamma_\text{y}} = 0.08$, $\mu_{\gamma_\text{xy}} = 0.006$, $x_0 = 1.4$, $A_{\text{t}} = 0.000055$, $B_{\text{t}} = 0.000235$, $\alpha_{\text{i}} = 0.62$ and $\alpha_{\text{o}} = 0.32$.

[1]The effect of random telegraph noise (RTN) on threshold voltage signal is less significant as compared to other noise components present in the flash channel [29], [30]

The final threshold voltage distribution can be computed as the convolution integral of initial voltage distribution function with RTN and data retention noise. Thus, we have

$$p_{\mathrm{s}_{11}}(v) = \frac{1}{\sigma_{\mathrm{s}_{11}}\sqrt{2\pi}} e^{-\frac{\left(v-\left(\tilde{V}_{\min}-\mu_{\mathrm{r}_{\mathrm{s}_{11}}}\right)\right)^2}{2\sigma^2_{\mathrm{s}_{11}}}} \tag{11}$$

$$\text{where } \sigma^2_{\mathrm{s}_{11}} = \sigma^2_{\mathrm{e}} + \sigma^2_{\mathrm{n}} + \sigma^2_{\mathrm{r}_{\mathrm{s}_{11}}}$$

$$p_{\mathrm{s}_{10}}(v) = \frac{1}{\Delta V_{\mathrm{pp}}}\left(\mathrm{erf}\left(\frac{V_1 + \Delta V_{\mathrm{pp}} - v - \mu_{\mathrm{r}_{\mathrm{s}_{10}}}}{\sqrt{2}\sigma_{\mathrm{s}_{10}}}\right)\right) - \frac{1}{\Delta V_{\mathrm{pp}}}\left(\mathrm{erf}\left(\frac{V_1 - v - \mu_{\mathrm{r}_{\mathrm{s}_{10}}}}{\sqrt{2}\sigma_{\mathrm{s}_{10}}}\right)\right) \tag{12}$$

$$\text{where } \sigma^2_{\mathrm{s}_{10}} = \sigma^2_{\mathrm{p}} + \sigma^2_{\mathrm{n}} + \sigma^2_{\mathrm{r}_{\mathrm{s}_{10}}}$$

$$p_{\mathrm{s}_{00}}(v) = \frac{1}{\Delta V_{\mathrm{pp}}}\left(\mathrm{erf}\left(\frac{V_2 + \Delta V_{\mathrm{pp}} - v - \mu_{\mathrm{r}_{\mathrm{s}_{00}}}}{\sqrt{2}\sigma_{\mathrm{s}_{00}}}\right)\right) - \frac{1}{\Delta V_{\mathrm{pp}}}\left(\mathrm{erf}\left(\frac{V_2 - v - \mu_{\mathrm{r}_{\mathrm{s}_{00}}}}{\sqrt{2}\sigma_{\mathrm{s}_{00}}}\right)\right) \tag{13}$$

$$\text{where } \sigma^2_{\mathrm{s}_{00}} = \sigma^2_{\mathrm{p}} + \sigma^2_{\mathrm{n}} + \sigma^2_{\mathrm{r}_{\mathrm{s}_{00}}}$$

$$p_{\mathrm{s}_{01}}(v) = \frac{1}{\Delta V_{\mathrm{pp}}}\left(\mathrm{erf}\left(\frac{V_{\max} + \Delta V_{\mathrm{pp}} - v - \mu_{\mathrm{r}_{\mathrm{s}_{01}}}}{\sqrt{2}\sigma_{\mathrm{s}_{01}}}\right)\right) - \frac{1}{\Delta V_{\mathrm{pp}}}\left(\mathrm{erf}\left(\frac{V_{\max} - v - \mu_{\mathrm{r}_{\mathrm{s}_{01}}}}{\sqrt{2}\sigma_{\mathrm{s}_{01}}}\right)\right) \tag{14}$$

$$\text{where } \sigma^2_{\mathrm{s}_{01}} = \sigma^2_{\mathrm{p}} + \sigma^2_{\mathrm{n}} + \sigma^2_{\mathrm{r}_{\mathrm{s}_{01}}}, \text{ and}$$

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-v^2} dv$$

III. Write-Voltage Signal Design

For a non-stationary flash memory channel, varying w.r.t. PE cycles and data retention time T, it is important to optimize and update the write-voltage levels for the optimal error-rate performance. For write-voltage optimization, it is assumed that the flash memory controller knows about the current PE count of individual memory blocks. The PE count history is mostly recorded inside the memory controller for performing the flash wear-levelling operation [32]. However, as the data retention time is generally difficult to predict at the time of cell programming, the retention time can either be set to zero ($T = 0$, representing the early retention time) or to some large value (e.g. $T = 1$ year, representing flash's end-of-retention). Given these two flash channel parameters, the four distribution functions (11)-(14) can be evaluated, paving the way for finding the optimal write-voltage levels. In other words, given $\tilde{V}_{\min}$ and $V_{\max}$ (based on the device physics), and the PE count, we next want to assign the intermediate write-voltage levels (V_1 and V_2). To answer this question, we formulate the probability of error expression P_{e} for the flash channel. This expression can be defined using the error probability of individual symbols, $P(e|s_{\mathrm{p}})$, given as

$$P_{\mathrm{e}} = \frac{1}{4}\{P(e|s_{11}) + P(e|s_{10}) + P(e|s_{00}) + P(e|s_{01})\} \tag{15}$$

where $1/4$ models the equi-probable input data symbols. To express $P(e|s_{\mathrm{p}})$, we need the decision boundaries R_1, R_2 and R_3 between adjacent distribution functions as shown in Fig. 1. Following this figure, we can define

$$P(e|s_{11}) = p_{\mathrm{s}_{11}}(v > R_1) \tag{16}$$

$$P(e|s_{10}) = p_{\mathrm{s}_{10}}(v < R_1) + p_{\mathrm{s}_{10}}(v > R_2) \tag{17}$$

$$P(e|s_{00}) = p_{\mathrm{s}_{00}}(v < R_2) + p_{\mathrm{s}_{00}}(v > R_3) \tag{18}$$

$$P(e|s_{01}) = p_{\mathrm{s}_{01}}(v < R_3) \tag{19}$$

These decision boundaries can also be considered as hard-decision levels on individual symbols. To get these decision boundaries, we equate the adjacent distribution functions and solve for the common intersecting point. Thus, for R_1, R_2 and R_3, we solve the following three equations

$$p_{\mathrm{s}_{11}}(v = R_1) = p_{\mathrm{s}_{10}}(v = R_1)$$

$$p_{\mathrm{s}_{10}}(v = R_2) = p_{\mathrm{s}_{00}}(v = R_2)$$

$$p_{\mathrm{s}_{00}}(v = R_3) = p_{\mathrm{s}_{01}}(v = R_3)$$

By simplifying these expressions we get $R_1 \in (\tilde{V}_{\min}, V_1)$, $R_2 \in (V_1, V_2)$ and $R_3 \in (V_2, V_{\max})$, respectively. Given the individual symbol error probabilities (16)-(19), we can re-write (15) in-terms of V_1, V_2, PE and T and optimize the desired parameters V_1 and V_2. In this section, we perform the voltage optimization for $T = 0$. The end-of-retention optimization will be discussed in the following section. Thus, instead of keeping the write-voltage levels fixed throughout the operational life-time of flash memory, we propose to adjust them according to the current channel condition such that the error probability is minimized. From the practical implementation perspective, the write-voltage levels can be pre-computed as a function of PE cycles and stored into a look-up table to preclude the run-time optimization complexity. To formulate the optimization problem, we define the objective function using (15) as

$$(V_1^*, V_2^*) = \min_{(V_1, V_2)} P_e(V_1, V_2, PE, T = 0) \tag{20}$$

This optimization function behaves as a convex function over V_1 and V_2 and can be solved by using any convex optimization technique. Since the write-voltage levels are optimized offline, the complexity of the chosen optimization algorithm is inconsequential. In this paper, we apply the gradient-descent (GD) [33] method to find the optimal V_1^* and V_2^* that yield the minimum P_{e}. The GD method minimizes the objective function P_{e} by iteratively solving the following equation

$$\begin{bmatrix} V_1^{(k+1)} \\ V_2^{(k+1)} \end{bmatrix} = \begin{bmatrix} V_1^{(k)} \\ V_2^{(k)} \end{bmatrix} - \eta \begin{bmatrix} \frac{\partial P_{\mathrm{e}}}{\partial V_1} \\ \frac{\partial P_{\mathrm{e}}}{\partial V_2} \end{bmatrix} P_{\mathrm{e}}\left(V_1^{(k)}, V_2^{(k)}, PE, T = 0\right) \tag{21}$$

where ∂ denotes the partial derivative, and k and η are the GD iteration count and step size, respectively.

The optimized write-voltage levels along with the corresponding minimum P_{e} values over different PE cycles are enumerated in Table. I. To make comparison between the proposed optimal write-voltage scheme and the conventional

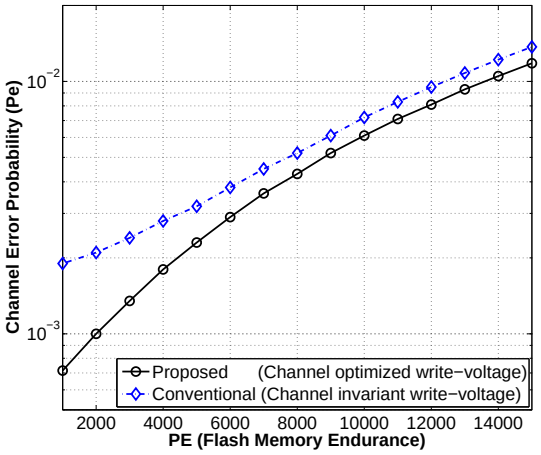

Fig. 3: Comparison of *channel error probability*, P_{e}, between the proposed write-voltage optimization scheme and the conventional channel invariant write-voltage scheme: *solid curve* (V_1^*, V_2^* from Table. I), *dotted curve* ($V_1 = 2.6, V_2 = 3.2$).

TABLE I: Optimized write-voltage levels over PE cycles.

PE	V_1^*	V_2^*	P_{e}
1000	2.77	3.35	7.15×10^{-4}
2000	2.75	3.34	0.0010
5000	2.69	3.31	0.0023
10000	2.61	3.27	0.0072
15000	2.55	3.24	0.0115

channel invariant (fixed) write-voltage scheme, we plot the P_{e} function as shown in Fig. 3. Here, the solid curve represents the minimized P_{e} values as given in Table I, whereas the dotted curve shows the P_{e} values computed against channel invariant write-voltage levels fixed at $V_1 = 2.6$ and $V_2 = 3.2$. As expected, the proposed scheme yields lower error probability, particularly at low-to-medium PE count. In addition to improved error performance, the optimal write-voltage levels also help to reduce the ECC decoding latency, as discussed in the subsequent section.

IV. Quantization Design for the Read-Voltage

In this section, we present a novel quantization scheme, using the voltage entropy function, to read flash memory cells. For 2−bit per cell flash memory, we require at least three quantization levels to detect four possible stored data symbols. However, when LDPC code is used as ECC for flash channel, we may need to perform high precision memory sensing for more accurate computation of log-likelihood-ratios (LLRs) for LDPC decoding. To achieve this, one straightforward approach is to set the quantization levels between $\tilde{V}_{\mathrm{min}}$ and V_{max} with equi-spaced separation D_{u}. Fig. 4 shows a pictorial representation of 6-level uniform quantization scheme, where each vertical dashed-line (R_1 to R_6) represent a particular quantization level. Since each flash symbol represents 2 bits, we compute the LLR values correspond to the most significant bit, L_{msb}, and the least significant bit, L_{lsb}, positions. Given the threshold voltage v, when $R_{n-1} \leqslant v \leqslant R_n$ for $n = 1, 2, 3, 4, 5, 6, 7$, where $R_0 = -\infty$ and $R_7 = +\infty$, we have

$$L_{\mathrm{msb}} = \log \frac{\int_{R_{n-1}}^{R_n} \{p_{\mathrm{s}_{00}}(v) + p_{\mathrm{s}_{01}}(v)\} dv}{\int_{R_{n-1}}^{R_n} \{p_{\mathrm{s}_{10}}(v) + p_{\mathrm{s}_{11}}(v)\} dv} \tag{22}$$

$$L_{\mathrm{lsb}} = \log \frac{\int_{R_{n-1}}^{R_n} \{p_{\mathrm{s}_{00}}(v) + p_{\mathrm{s}_{10}}(v)\} dv}{\int_{R_{n-1}}^{R_n} \{p_{\mathrm{s}_{01}}(v) + p_{\mathrm{s}_{11}}(v)\} dv} \tag{23}$$

However, for a given memory sensing precision (e.g. 6-level, 9-level, 12-level), the uniform quantization scheme is shown to be ineffective as compared to the non-uniform quantization schemes [14]. The objective of a non-uniform quantization scheme is to set the memory sensing levels closer to the erasure (intersection) regions where adjacent distribution functions are overlapped. Since the symbol error probability is dominant in this region, it requires more accurate memory sensing. Furthermore, for a non-uniform quantization scheme, it is crucial to identify the optimum width of erasure region so that the resultant soft-information (LLR values) achieves the best decoder error performance. For this purpose, we first define the entropy of cell's threshold voltage, $H(v)$, given by

$$H(v) = \sum_i \left[\frac{p_{\mathrm{s}_i}(v)}{\sum_i p_{\mathrm{s}_i}(v)} \log_2 \left(\frac{\sum_i p_{\mathrm{s}_i}(v)}{p_{\mathrm{s}_i}(v)} \right) \right] \tag{24}$$

where $i \in \{11, 10, 00, 01\}$. An example of voltage entropy function is illustrated in Fig. 5. It can be observed that the voltage entropy is only dominant within the erasure region, denoted as *high-entropy-region*. This should be the targeted memory sensing region as it incurs high error probability. Therefore, we set the quantization levels closer to the erasure region, instead of setting them uniformly from $\tilde{V}_{\mathrm{min}}$ to V_{max}. With 3 erasure regions in-place, we require atleast 6 quantization levels in order to enclose them within the quantization boundary. To this end, we define a parameter $\theta \in [0, 1]$ to select the width of each erasure region. To be precise, we set each quantization level such that

$$H(R_n) = \theta \tag{25}$$

for $n = 1, 2, 3, 4, 5, 6$. In other words, the memory sensing levels are designed where the voltage entropy is equal to θ. By varying the entropy parameter θ, we can obtain the desired width for erasure regions. We refer to this memory sensing scheme as *entropy-based* quantization scheme. Besides, another well-known non-uniform quantization scheme for MLC flash is reported in [17], where the quantization levels are obtained by maximizing the mutual-information (MMI) between the input and output of flash channel. This scheme involves solving a multi-variable optimization problem to get the quantization levels. However, with MMI quantization, it is not possible to select the width of erasure regions. This constraint affects the resultant error-rate performance of LDPC

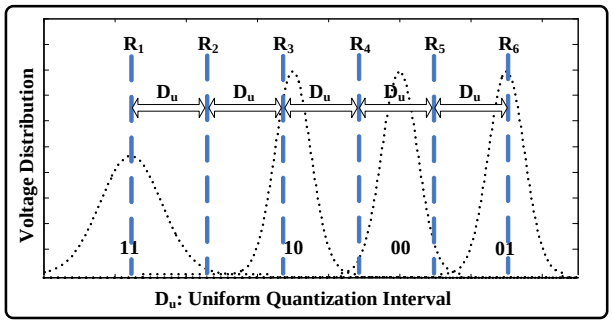

Fig. 4: Illustration of *uniform* interval D_{u} based quantization scheme: vertical lines represent quantization levels (R_1 to R_6).

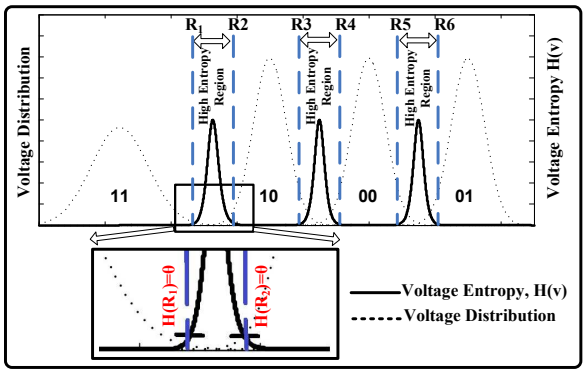

Fig. 5: Illustration of *entropy-based* quantization scheme for 2-bit per cell flash memory: quantization levels (R_1 to R_6) are placed closer to the erasure (overlapping) regions.

decoder, which is further discussed in the following section. Since both proposed and MMI schemes use an entropy like function to optimize the quantization levels, we mainly treat MMI quantization as the benchmark for comparison.

To observe the error-rate performance, we simulate two binary LDPC codes, referred to as *4K-code* and *8K-code*, over the model-based 2-bit per cell flash memory. The 4K-code is an irregular LDPC code with input and output block-length (frame size) of 4096 and 4544 bits, respectively, and code-rate of 0.90. The degree distribution of this code is as follows:

$$\lambda(x) = 0.0682x + 0.1822x^2 + 0.1329x^3 + 0.6167x^4$$
$$\rho(x) = 0.22x^{38} + 0.78x^{39}$$

where $\lambda(x)$ and $\rho(x)$ are the variable-node and check-node degree distribution pairs, respectively, optimized through density-evolution [34]. The 8K-code is chosen as a regular LDPC code with uniform column-weight of 4. It has input and output block-length of 7360 and 8000 bits, respectively, and code-rate of 0.92. Both LDPC codes are constructed using progressive-edge-growth (PEG) algorithm [35], and decoded using column-weight based shuffled belief-propagation decoder [36], with maximum iteration count, $I_{\max}$, set to 25. We apply the proposed entropy-based quantization scheme (6-level) and compare with MMI (6-level) [17] and uniform (12-level) quantization schemes. The entropy parameter is set to $\theta = 0.35$ as it leads to optimum error-rate performance.

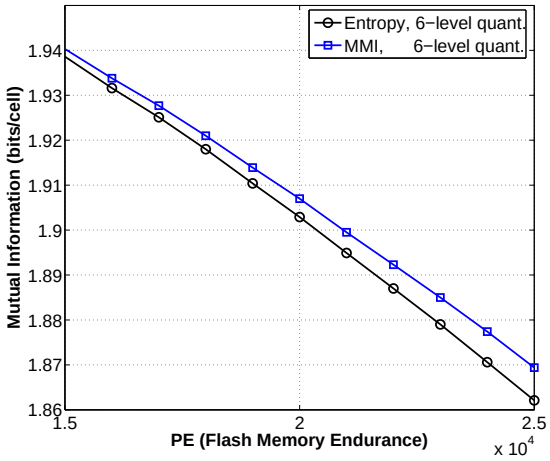

Fig. 6: Comparison of *mutual information* obtained through 6-level entropy (proposed) and MMI [17] quantization schemes.

We first notice that, contrary to MMI quantization, the entropy-based scheme does not maximize the mutual information as shown in Fig. 6. Next, we plot the frame-error-rate (FER) curves for LDPC 4K-code and 8K-code with retention time set to zero ($T = 0$) as shown in Fig. 7. In this figure, we show the error performance of MMI and uniform quantization schemes by using both channel invariant (conventional) and channel optimized (proposed) write-voltage levels. We observe that the proposed entropy-based quantization outperforms the former schemes. Furthermore, as stated earlier, the uniform quantization scheme is not very effective as the 12-level quantization is, surprisingly, worse than the 6-level non-uniform quantization schemes.

In order to simulate the flash memory under data retention scenario, we need to estimate the voltage distribution functions prior to memory sensing operation. This is required because the retention noise shifts the voltage distribution functions over increasing retention time T. In this work, we rely upon the prior-art schemes [37], [38] to estimate the shifted distribution functions. Based on that, we compute new set of entropy-based quantization levels and derive the corresponding LLRs using (22) and (23). In Fig. 8, we illustrate the frame-error-rate (FER) performance of LDPC 4K-code, where the retention time is set to 6 months, 12 months and 24 months, respectively. It is evident that the proposed entropy-based quantization scheme has superior error-rate performance as compared to the MMI scheme, substantiating the effectiveness of the proposed quantization under data retention scenario.

It should be noted that, in this paper, we strive to optimize the 6-level quantization scheme. This is because the channel capacity with 6-level quantization registers a big jump from 3-level quantization (hard decision), while further increase in the quantization levels registers diminishing returns towards the infinite-precision limit, as shown in Fig.9.

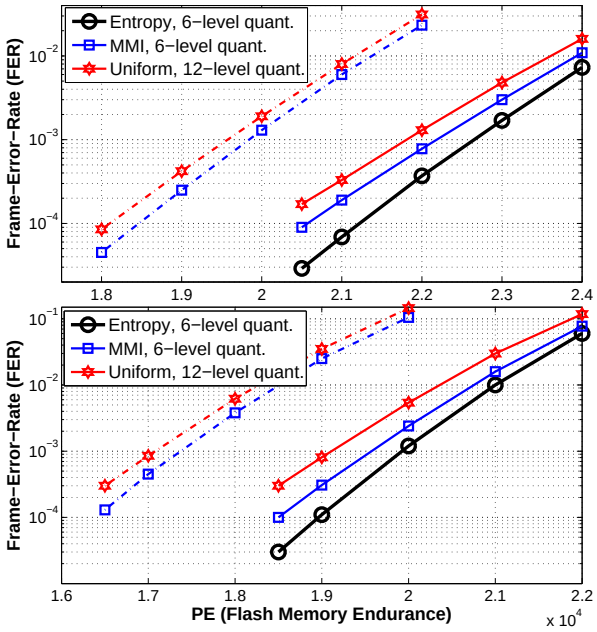

Fig. 7: *Frame-error-rate* (FER) performance of LDPC 4K-code (top) and LDPC 8K-code (bottom) using entropy (proposed), MMI [17] and uniform quantization schemes: *solid curves* are plotted for optimized write-voltage levels and *dotted curves* are plotted for channel invariant write-voltage levels.

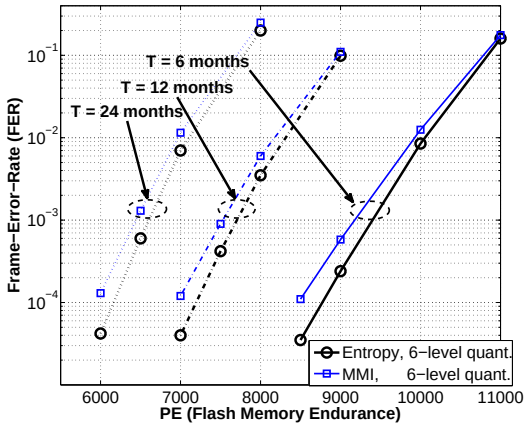

Fig. 8: *Frame-error-rate* (FER) performance of LDPC 4K-code using entropy (proposed) and MMI [17] quantization schemes, setting retention time (T) at 6, 12 and 24 months.

V. Controlled Erasure Decoding using Entropy based Quantization

In this section, we investigate the reason behind the improvement in error performance achieved by using the proposed entropy-based quantization scheme over the MMI quantization scheme [17]. We first draw attention to the fact that the entropy-based quantization levels (given by the optimal value of θ) tend to produce wider erasure regions than the MMI quantization levels, as depicted in Fig. 10. In this figure, we observe that the threshold voltage range is split into 3 erasure regions $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_3$, and 4 data regions $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ and $\mathcal{D}_4$,

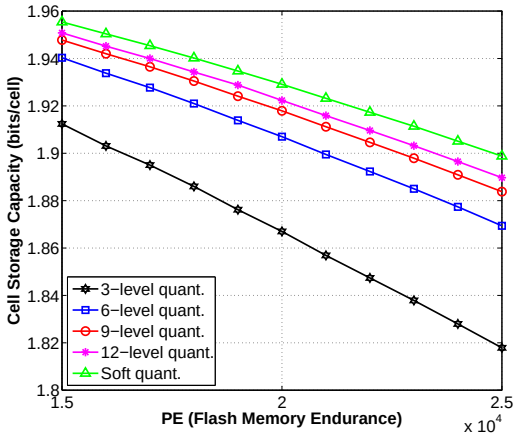

Fig. 9: Plot of *cell storage capacity* using 3-level, 6-level, 9-level, 12-level and *soft* quantization schemes.

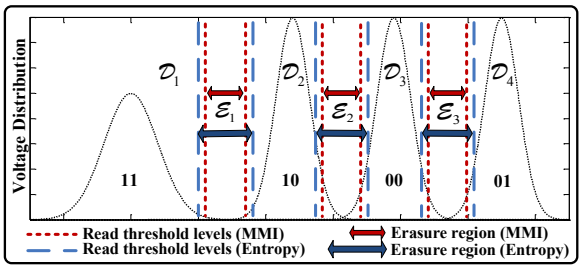

Fig. 10: Comparison of 6-level memory sensing between the entropy (proposed) and MMI [17] quantization schemes: entropy quantization levels contribute to wider erasure regions.

respectively. Here, it must be realized that adjusting the width of erasure regions affects the error performance of the code, as shown in Fig. 11. This plot shows the error performance of LDPC 4K-code over a range of θ values for PE = 21K. The width of erasure regions corresponding to different θ values are shown in Table. II. Observing Fig. 11 and Table. II reveals that if the erasure region is too narrow, the decoder will not be able to determine sufficient erasures; while if the erasure region is too wide, it will get too many erased symbols which exceed the code's error correcting capability; and the entropy parameter θ allows the erasure region width to be optimized. With MMI quantization, it is not possible to alter the width of erasure region as it is optimized (fixed) over a given channel (for a fixed channel noise), irrespective of the type of code and decoding algorithm used. Fig. 11 also marks the entropy parameter value corresponding to the MMI quantization levels, which clearly does not coincide with the minimum error-rate point. This is because the MMI quantization optimization attempts to achieve zero error probability while assuming infinite channel code length, but in practical systems this assumption is violated as only finite block-length codes can be adopted. On the contrary, in entropy-based quantization, changing the value of θ influences the symbol error probability, P_{error}, and the symbol erasure probability, P_{erasure}, of the flash channel, and consequently alters the magnitude of input LLRs

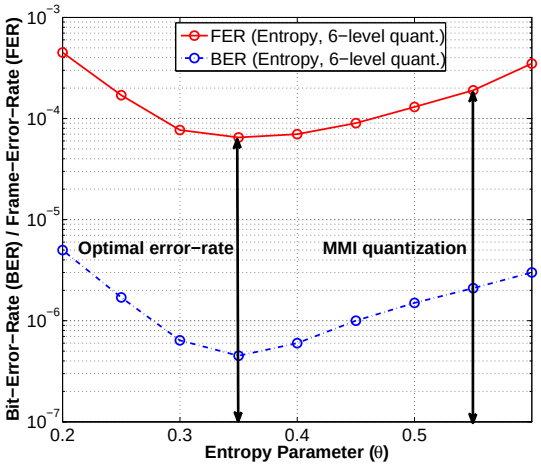

Fig. 11: *BER/FER* performance of LDPC 4K-code over varying entropy parameter θ at PE = 21K: values of θ corresponding to optimal error-rate performance and MMI quantization levels are marked with vertical arrows.

used for BP decoding. Here, P_{error} refers to the uncoded error probability computed in 4 data regions and P_{erasure} refers to the erasure probability computed in 3 erasure regions. These two quantities are mathematically written as

$$\begin{aligned} P_{\text{error}} = &\int_{v\in\mathcal{D}_1} \left(p_{\text{s}_{10}}(v) + p_{\text{s}_{00}}(v) + p_{\text{s}_{01}}(v)\right) dv \\ &+ \int_{v\in\mathcal{D}_2} \left(p_{\text{s}_{11}}(v) + p_{\text{s}_{00}}(v) + p_{\text{s}_{01}}(v)\right) dv \\ &+ \int_{v\in\mathcal{D}_3} \left(p_{\text{s}_{11}}(v) + p_{\text{s}_{10}}(v) + p_{\text{s}_{01}}(v)\right) dv \\ &+ \int_{v\in\mathcal{D}_4} \left(p_{\text{s}_{11}}(v) + p_{\text{s}_{10}}(v) + p_{\text{s}_{00}}(v)\right) dv \end{aligned} \tag{26}$$

$$\begin{aligned} P_{\text{erasure}} = &\int_{v\in\mathcal{E}_1} \left(p_{\text{s}_{11}}(v) + p_{\text{s}_{10}}(v) + p_{\text{s}_{00}}(v) + p_{\text{s}_{01}}(v)\right) dv \\ &+ \int_{v\in\mathcal{E}_2} \left(p_{\text{s}_{11}}(v) + p_{\text{s}_{10}}(v) + p_{\text{s}_{00}}(v) + p_{\text{s}_{01}}(v)\right) dv \\ &+ \int_{v\in\mathcal{E}_3} \left(p_{\text{s}_{11}}(v) + p_{\text{s}_{10}}(v) + p_{\text{s}_{00}}(v) + p_{\text{s}_{01}}(v)\right) dv \end{aligned} \tag{27}$$

We evaluate and plot these two expressions for different PE cycles in Fig. 12. In this figure, compared to MMI, the proposed entropy-based quantization reduces the symbol error probability by having a higher erasure probability. It should be noted that the LLR values associated with erasure regions are the least reliable LLRs and therefore reset to zero (erased). To be precise, the LSB LLRs correspond to $\mathcal{E}_1$ and $\mathcal{E}_3$, and the MSB LLRs correspond to $\mathcal{E}_2$ are erased. Thus, with entropy-based quantization scheme, we perform a controlled erasure decoding by means of optimizing the parameter θ, in such a way that some LLR values are erased to ensure that the symbol error probability is reduced. This adjustment between the symbol error and symbol erasure probability helps the entropy scheme to perform better than the MMI scheme.

TABLE II: Width of erasure regions $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_3$ over varying entropy parameter θ.

θ	$\mathcal{E}_1$	$\mathcal{E}_2$	$\mathcal{E}_2$
0.20	0.336	0.221	0.220
0.25	0.303	0.202	0.202
0.30	0.277	0.185	0.184
0.35	**0.253**	**0.171**	**0.170**
0.40	0.233	0.158	0.156
0.45	0.214	0.146	0.144
0.50	0.197	0.135	0.133
0.55	0.181	0.124	0.122
0.60	0.166	0.114	0.112

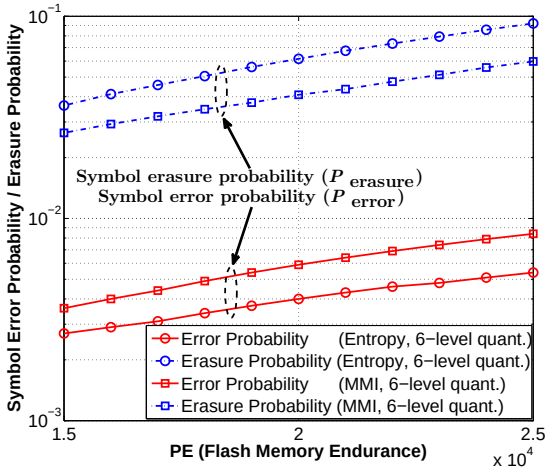

Fig. 12: Comparison of *symbol error probability*, P_{error}, and *symbol erasure probability*, P_{erasure}, between the entropy (proposed) and MMI [17] quantization schemes: entropy quantization produces lower P_{error} at the cost of higher P_{erasure}.

As a future work, it is of great interest to analytically evaluate the optimum value of θ for which the decoded error probability is minimized. Apart from entropy function, it is compelling to investigate other non-linear functions to see if the error performance can be further improved.

VI. Write-Voltage Optimization for Faster Decoding Convergence

In the previous section, we mainly analyze the improvement in error-rate performance that comes as a result of using optimal write-voltage signals, as shown in Fig. 3 and Fig. 7. It should be noted that the need for write-voltage optimization is not critical in the initial state of flash memory as compared to its end-of-life (EOL). This is because the error-rate in the initial state typically does not exceed the error correction capability of ECC decoder. However, as the LDPC code is becoming the mainstream ECC in flash memory controller, its long decoding latency starts to deteriorate the system performance. In this situation, the proposed optimization method to reduce the error-rate, and consequently to reduce the decoding latency, becomes important. It other words, we can improve the decoding convergence in the initial state of flash memory

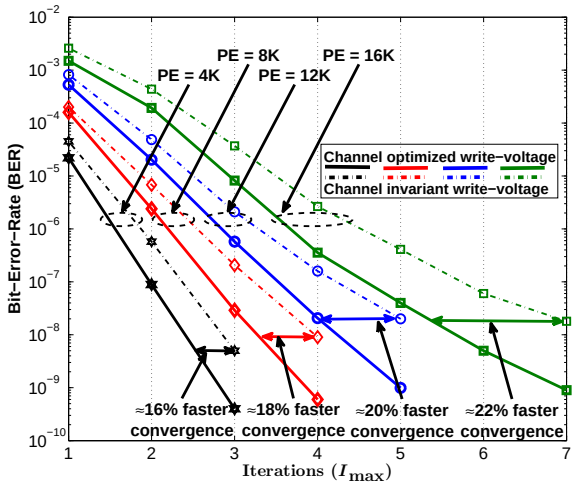

Fig. 13: *Bit-error-rate* (BER) performance of LDPC 4K-code plotted versus decoding iterations ($I_{\max}$) for different PE cycles: *solid* and *dotted* curves represent channel optimized and channel invariant (fixed) write-voltage schemes, respectively.

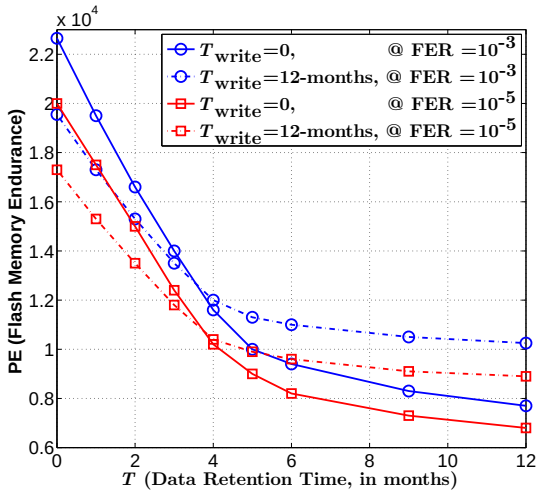

Fig. 14: Comparison of *flash memory endurance* between optimizing write-voltage for early retention, $T_{\text{write}} = 0$ (*solid curves*), and end-of-retention (EOR), $T_{\text{write}} = 12$-months (*dotted curves*), respectively, keeping the FER fixed at 10^{-3} and 10^{-5}: using LDPC 4K-code and 6-level entropy quantization.

by optimizing the write-voltage levels. It should be noted that the total read latency of NAND flash memory based devices is composed of firmware processing, memory sensing, and DMA (data transfer from controller to NAND flash memory) times, and the most time consuming process is the memory sensing operation. The LDPC decoding latency in included in the DMA time.

In Fig. 13, we plot the BER curves versus the maximum iteration count, $I_{\max}$, for different PE cycles. It can be observed that the decoding latency is reduced by up to 16% in the initial state, even though the BER at these early states are low and typically have enough margins from ECC failures, and up to 22% in the EOL. Therefore, in comparison with conventional flash memory system where the write-voltage levels are fixed, our approach is more effective in-terms of error performance and decoding latency for both early and late stages of flash memory usage.

VII. Write-Voltage Optimization for Flash End-Of-Retention (EOR) Time

Previously, we performed the write-voltage optimization for the early retention time of flash memory, by setting the retention time to zero. However, in this section, we address the write-voltage scheme optimized for the flash memory's end-of-retention (EOR) time, and compare between the two voltage design schemes. In this direction, we set the retention time to some large value and then minimize the objective function (20). As a case study, we set the retention time to 12 months and optimize the write-voltage levels over varying PE cycles. The optimization for EOR is recommended if the flash data is expected to be stored for a long period of time without reading/re-writing. We use $T_{\text{write}} = 0$ and $T_{\text{write}} = 12$-months to distinguish between optimizing the write-voltage schemes for early retention and end-of-retention times, respectively.

In Fig. 14, we plot the flash memory endurance (PE cycles) versus the data retention time (T) for the two voltage design schemes, $T_{\text{write}} = 0$ and $T_{\text{write}} = 12$-months, keeping the FER fixed at 10^{-3} and 10^{-5}. We observe that if the data retention time is relatively short, under 4 months, the write-voltage scheme optimized for early retention time outperforms the EOR optimization scheme by providing longer flash memory endurance. Conversely, as the data retention time increases, the write-voltage scheme optimized for EOR time yields better endurance performance. In summary, the problem of write-voltage optimization for MLC flash is a design trade-off between the achievable endurance performance and the expected retention time of stored flash data.

VIII. Minimum Memory Sensing Precision for LDPC Decoder Convergence

Until now, we perform simulations using 6-level memory sensing precision. Since the flash memory channel is non-stationary, even the number of quantization levels should not be fixed but should instead be optimized in conjunction with soft LDPC decoding. For instance, it is intuitive to expect that when flash memory is relatively fresh and has gone through small number of PE cycles, we can use fewer quantization levels, say 3-level. However, as the channel noise variance increases due to larger number of PE cycle, we must switch to higher quantization levels, e.g. 6-level. This adaptive memory precision approach has been reported in [20] in which the author has used the minimum quantization levels (1-level for $1-$bit per cell, 3-levels for $2-$bit per cell flash memory, and so on) to initiate the LDPC decoder and progressively increased the memory precision if the lower-precision sensing fails to produce successful LDPC code-word. However, this method is not optimal in terms of decoding latency since the flash read-out data is decoded for different quantization levels until the code-word is successfully produced. Ideally,

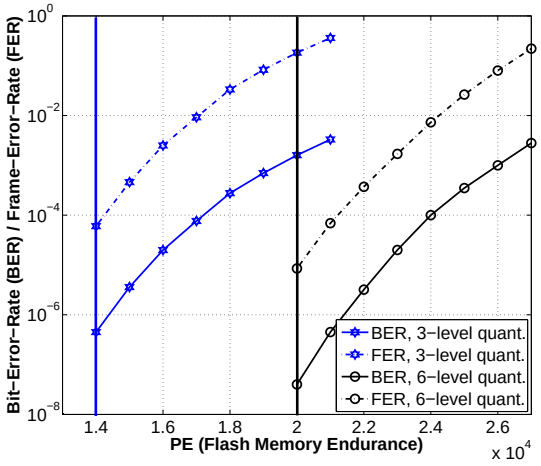

Fig. 15: FER (*dotted curves*) and BER (*solid curves*) performance along with EXIT limit (*vertical lines*) of LDPC 4K-code for 3-level and 6-level quantization schemes.

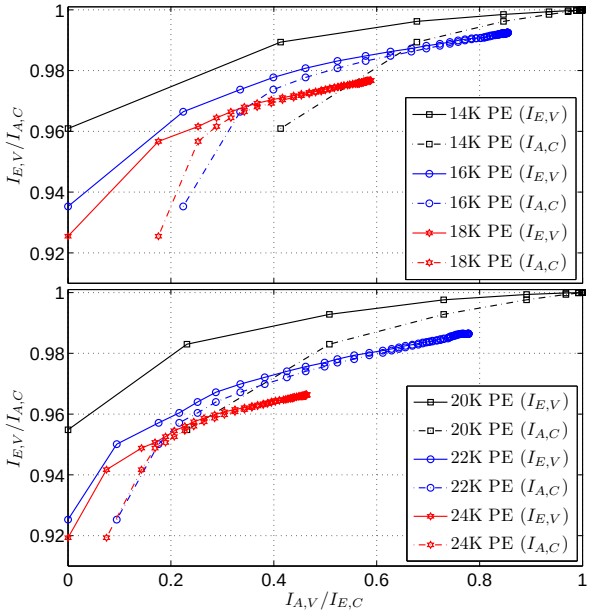

Fig. 16: *EXIT* curves for LDPC 4K-code using 3-level (upper plot) and 6-level (lower plot) entropy quantization schemes over different PE cycles.

we should not always start the decoding process from the minimum quantization levels, but should select the memory sensing precision based on the knowledge of the current flash channel status (PE count).

In this direction, we propose to select the minimum required number of quantization levels based on the extrinsic information transfer (EXIT) curves [39]. For a given LDPC code, we plot the EXIT diagram using Monte-Carlo simulations for different quantization levels over varying PE cycles and select the smallest number of read levels which successfully transfer the mutual-information curve. This EXIT curve based quantization selection approach seems to be more reliable as the simulations take into account the effect of LDPC finite block-length, providing more accurate prediction of the error-rate performance. Alternatively, relying on the channel capacity curve to choose the number of quantization levels may not be accurate enough because of its infinite block-length assumption. In Fig. 16, using the same LDPC 4K-code, we plot the EXIT curves for 3-level ($\theta = 1$) and 6-level ($\theta = 0.35$) quantization schemes for different PE cycles, keeping the retention time fixed at $T = 0$. For 3-level quantization, we may notice that beyond 14K PE cycles, the mutual-information curve is not successfully transferred. At this point, we observe in Fig. 15 that the BER is around 10^{-6}. Therefore, if we want to operate the flash memory beyond 14K PE cycles, we must switch to higher precision memory sensing. This can be verified from Fig. 16, which shows that 6-level quantization can be employed between 14K to 20K PE cycles, beyond which we again require higher quantization levels. Thus, EXIT analysis can be used as a reliable tool to choose the required number of read-voltage levels.

IX. Conclusion

This paper investigates the optimization of read and write voltage signals for MLC NAND flash memory. Based on a non-stationary flash channel model, considering programming noise, random telegraph noise, cell-to-cell interference and data retention noise, the write-voltage levels are optimized by minimizing the channel error probability. The trade-off between write-voltage optimization based on early retention time versus end-of-life retention time of the stored data is discussed. The proposed write-voltage optimization scheme is validated to have superior error-rate performance over the conventional channel-invariant write-voltage scheme.

Besides, a novel read-voltage quantization scheme based on the voltage entropy function is proposed to find the right balance between the flash channel symbol error and symbol erasure probability through controlled erasure decoding. This quantization scheme enables the LDPC decoder to achieve soft BP decoding performance improvement over prior-art quantization schemes. By virtue of both write and read voltage optimization, the overall error-rate performance and decoding convergence are improved. Finally, a method to minimize the number of quantization levels based on EXIT curve is proposed to enable the flash memory controller to reduce the memory sensing latency while guaranteeing the LDPC decoder convergence.

References

[1] K. Kim, "Future memory technology: challenges and opportunities," in *Proc. Int. Symp. VLSI Technol., Syst. Appl.*, San Jose, CA, 2008, pp. 5–9.

[2] L. M. Grupp, J. D. Davis, and S. Swanson, "The bleak future of NAND flash memory," in *Proc. 10th USENIX Conf. File and Storage Technol.*, ser. FAST'12, Berkeley, CA, USA, 2012, pp. 2–2.

[3] Q. Li, A. Jiang, and E. Haratsch, "Noise modeling and capacity analysis for NAND flash memories," in *IEEE Int. Symp. Inf. Theory*, Honolulu, HI, June 2014, pp. 2262–2266.

[4] K. Zhao *et al.*, "LDPC-in-SSD: Making advanced error correction codes work effectively in solid state drives," in *Proc. 11th USENIX Conf. File and Storage Technol.*, ser. FAST'13, San Jose, CA, 2013, pp. 243–256.

[5] C. Aslam, Y. Guan, and K. Cai, "Non-binary LDPC code with multiple memory reads for multi-level-cell (MLC) flash," in *Proc. Int. Conf. APSIPA*, Dec 2014, pp. 1–9.

[6] ——, "Dynamic write-level and read-level signal design for MLC NAND flash memory," in *Proc. 9th Int. Symp. Commun. Syst., Netw., Digit. Signal Process.*, Manchester, UK, July 2014, pp. 336–341.

[7] Y. Kim *et al.*, "Verify level control criteria for multi-level cell flash memories and their applications," *EURASIP J. Adv. Sig. Proc.*, vol. 2012, no. 1, p. 196, 2012.

[8] Y. Cassuto *et al.*, "Codes for multi-level flash memories: Correcting asymmetric limited-magnitude errors," in *Proc. IEEE Int. Symp. Inf. Theory*, Nice, France, June 2007, pp. 1176–1180.

[9] A. Berman and Y. Birk, "Constrained flash memory programming," in *Proc. IEEE Int. Symp. Inf. Theory*, Saint Petersburg, Russia, July 2011, pp. 2128–2132.

[10] J. Anxiao *et al.*, "Rank modulation for flash memories," *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2659–2673, June 2009.

[11] A. Jiang, H. Li, and J. Bruck, "On the capacity and programming of flash memories," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1549–1564, March 2012.

[12] A. Jiang and H. Li, "Optimized cell programming for flash memories," in *Proc. IEEE PACRIM*, Victoria, B.C., Canada, Aug 2009, pp. 914–919.

[13] M. Qin, E. Yaakobi, and P. Siegel, "Optimized cell programming for flash memories with quantizers," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2780–2795, May 2014.

[14] G. Dong, N. Xie, and T. Zhang, "On the use of soft-decision error-correction codes in NAND flash memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 2, pp. 429–439, Feb 2011.

[15] F. Sala, R. Gabrys, and L. Dolecek, "Dynamic threshold schemes for multi-level non-volatile memories," *IEEE Trans. Commun.*, vol. 61, no. 7, pp. 2624–2634, July 2013.

[16] Y. Cai *et al.*, "Data retention in MLC NAND flash memory: Characterization, optimization, and recovery," in *Proc. IEEE 21st Int. Symp. High Performance Computer Architecture*, San Francisco, CA, Feb 2015, pp. 551–563.

[17] W. Jiadong *et al.*, "Enhanced precision through multiple reads for LDPC decoding in flash memories," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 880–891, May 2014.

[18] L. P. R. Z. Xueqiang Wang, Guiqiang Dong, "Error correction codes and signal processing in flash memory," in *Flash Memories*, P. I. Stievano, Ed. InTech, 2011.

[19] D. Guiqiang *et al.*, "Estimating information-theoretical NAND flash memory storage capacity and its implication to memory system design space exploration," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 9, pp. 1705–1714, Sept 2012.

[20] G. Dong, N. Xie, and T. Zhang, "Enabling NAND flash memory use soft-decision error correction codes at minimal read latency overhead," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 9, pp. 2412–2421, Sept 2013.

[21] T.-Y. Chen, A. Williamson, and R. Wesel, "Increasing flash memory lifetime by dynamic voltage allocation for constant mutual information," in *Proc. Inf. Theory and Appl. Workshop*, San Diego, CA, Feb 2014, pp. 1–5.

[22] G. Dong, Y. Pan, and T. Zhang, "Using lifetime-aware progressive programming to improve SLC NAND flash memory write endurance," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 6, pp. 1270–1280, June 2014.

[23] H. Wang, T.-Y. Chen, and R. Wesel, "Histogram-based flash channel estimation," in *Proc. IEEE Int. Conf. Commun.*, June 2015, pp. 283–288.

[24] X. Quan *et al.*, "Modelling and characterization of NAND flash memory channels," *Measurement*, vol. 70, pp. 225 – 231, 2015.

[25] K. Takeuchi, T. Tanaka, and H. Nakamura, "A double-level-vth select gate array architecture for multilevel NAND flash memories," *IEEE J. Solid-State Circuits*, vol. 31, no. 4, pp. 602–609, 1996.

[26] S. Kang-Deog *et al.*, "A 3.3 V 32 Mb NAND flash memory with incremental step pulse programming scheme," *IEEE J. Solid-State Circuits*, vol. 30, no. 11, pp. 1149–1156, 1995.

[27] J.-D. Lee, S.-H. Hur, and J.-D. Choi, "Effects of floating-gate interference on NAND flash memory cell operation," *IEEE Electron Device Lett.*, vol. 23, no. 5, pp. 264–266, May 2002.

[28] G. Dong, S. Li, and T. Zhang, "Using data postcompensation and predistortion to tolerate cell-to-cell interference in MLC NAND flash memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 10, pp. 2718–2728, Oct 2010.

[29] Y. Cai *et al.*, "Error patterns in MLC NAND flash memory: Measurement, characterization, and analysis," in *Proc. Design, Autom., Test Eur.*, ser. DATE '12, San Jose, CA, USA, 2012, pp. 521–526.

[30] ——, "Error analysis and retention-aware error management for NAND flash memory," *Intel Technology Journal (ITJ)*, vol. 17, no. 1, p. 140, 2013.

[31] C. M. Compagnoni *et al.*, "Random telegraph noise effect on the programmed threshold-voltage distribution of flash memories," *IEEE Electron Device Lett.*, vol. 30, no. 9, pp. 984–986, 2009.

[32] K. M. Lofgren *et al.*, "Wear leveling techniques for flash EEPROM systems," Feb. 1 2005, US Patent 6,850,443.

[33] J. A. Snyman, *Practical mathematical optimization : An introduction to basic optimization theory and classical and new gradient-based algorithms*, ser. Applied optimization. New York: Springer, 2005.

[34] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb 2001.

[35] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan 2005.

[36] C. Aslam, Y. Guan, and K. Cai, "Improving the belief-propagation convergence of irregular LDPC codes using column-weight based scheduling," *IEEE Commun. Lett.*, vol. 19, no. 8, pp. 1283–1286, Aug 2015.

[37] D. hwan Lee and W. Sung, "Estimation of NAND flash memory threshold voltage distribution for optimum soft-decision error correction," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 440–449, Jan 2013.

[38] Y. Cai *et al.*, "Threshold voltage distribution in MLC NAND flash memory: Characterization, analysis, and modeling," in *Proc. Design, Autom., Test Eur.*, Grenoble, France, March 2013, pp. 1285–1290.

[39] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: Model and erasure channel properties," *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2657–2673, Nov 2004.

Chaudhry Adnan Aslam received his B.E. degree from Mehran University of Engineering and Technology Pakistan and M.S. degree from the University of Southern California (USC) Los Angeles, CA. in 2005 and 2009, respectively. He is currently pursuing the Ph.D. degree from the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. His research interests include coding theory and signal processing for data storage and wireless communications systems.

Yong Liang Guan obtained his Ph.D. from the Imperial College of London, UK, and Bachelor of Engineering with first class honors from the National University of Singapore. He is now an Associate Professor at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests broadly include modulation, coding and signal processing for communication systems, and information security systems. His homepage is at http://www3.ntu.edu.sg/home/eylguan/index.htm.

Kui Cai received B.E. degree in information and control engineering from Shanghai Jiao Tong University, Shanghai, China, M.Eng degree in electrical engineering from National University of Singapore, and joint Ph.D. degree in electrical engineering from Technical University of Eindhoven, The Netherlands, and National University of Singapore. Currently she is an Associate Professor with Singapore University of Technology and Design (SUTD). Before joining SUTD, she had been with Data Storage Institute (DSI), Singapore, since 1999, where she was the Program Leader of non-volatile memory (NVM) coding and signal processing. Cai Kui is a senior member of IEEE and the Vice-Chair (Academia) of IEEE Communications Society, Data Storage Technical Committee (DSTC). She is the recipient of 2008 IEEE Communications Society Best Paper Award in Coding and Signal Processing for Data Storage. Her research interests include coding theory, communication theory, and signal processing for various data storage systems and digital communications.